UNCLASSIFIED

AD NUMBER

AD911417

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; MAR 1973. Other requests shall be referred to Army Electronics Command, For Monmouth, NJ 07703.

AUTHORITY

USAEC ltr, 4 Mar 1974

THIS PAGE IS UNCLASSIFIED

AD 911417

AD

Research and Development Technical Report
ECOM-0292-6

ANALYTIC MATHEMATICAL MODELS OF TACTICAL MILITARY COMMUNICATIONS CHANNELS

QUARTERLY REPORT

MAY, 1973

R.T. CHIEN
C.L. CHEN
S. TSAI

DISTRIBUTION STATEMENT
Distribution limited to U.S. Government agencies only; Test and Evaluation; March 73. Other requests for this document must be referred to Commanding General U.S. Army Electronics Command, ATTN: AMSEL-NL-R-2, Fort Monmouth, New Jersey 07703

ECOM

UNITED STATES ARMY ELECTRONICS COMMAND · FORT MONMOUTH, N.J.

Contract DAAB07-71-C-0292
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

TR ECOM-0292-6
May 1973

Reports Control Symbol
OSD-1366

ANALYTIC MATHEMATICAL MODELS OF TACTICAL MILITARY COMMUNICATIONS CHANNELS

SIXTH QUARTERLY PROGRESS REPORT

1 October 1972 - 31 December 1972

Contract No. DAAB07-71-C-0292

DA Project No. 1S6.62701.A327.06.07

DISTRIBUTION STATEMENT

Distribution limited to U.S. Government agencies only; Test and Evaluation; March 1973. Other requests for this document must be referred to Commanding General, U.S. Army Electronics Command, ATTN: AMSEL-NL-R-2, Fort Monmouth, New Jersey 07703

Prepared by

R. T. Chien
C. L. Chen
S. Tsai

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

For

U. S. ARMY ELECTRONICS COMMAND, FORT MONMOUTH, N. J.

ABSTRACT

The Viterbi decoding algorithm yields minimum probability of error when applied to a memoryless channel provided that all input sequences are equally likely. In this report, the algorithm was generalized for application to channels with finite memory and it was shown that the generalized algorithm is also maximum-likelihood decoding. It was also shown that the generalized Viterbi algorithm on a simple memory channel performs better than the original Viterbi algorithm with the same decoding complexity.

The M-state Markov model was reviewed in this report. The process of identifying the parameters of the M-state model from the coefficients A_i and $A_i(n_i,n_{i+1})$ of the gap model was determined to be more complicated than was anticipated. As an alternative, the simple partitioned Markov model was examined to determine the effect of the second order statistics, namely the interdependence of the gaps, on the error burst distribution. An alternative definition of the burst was adopted to speed up this investigation. The difference or similarity between these two definitions will be determined.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

SUMMARY

The Viterbi decoding algorithm yields minimum probability of error when applied to a memoryless channel provided that all input sequences are equally likely. In this report, the algorithm was generalized for application to channels with finite memory and it was shown that the generalized algorithm is also maximum-likelihood decoding. It was also shown that the generalized Viterbi algorithm on a simple memory channel performs better than the original Viterbi algorithm with the same decoding complexity.

The M-state Markov model was reviewed in this report. The process of identifying the parameters of the M-state model from the coefficients A_i and $A_i(n_i,n_{i+1})$ of the gap model was determined to be more complicated than was anticipated. As an alternative, the simple partitioned Markov model was examined to determine the effect of the second order statistics, namely the interdependence of the gaps, on the error burst distribution. An alternative definition of the burst was adopted to speed up this investigation. The difference or similarity between these two definitions will be determined.

SECTION 2

VITERBI DECODING ALGORITHM

2.1. INTRODUCTION

In 1955 Elias [1] introduced a new class of codes, called convolutional codes (sometimes called recurrent codes), which has become an important alternative to the block coding scheme. Unlike the block coding scheme which divides a string of information digits into blocks of k digits each and encodes these blocks into blocks of codewords of n digits each, the convolutional coding scheme takes a string of information digits of arbitrary length (can be semi-infinite) and encodes it into a single string of coded digits. More on the structure of the convolutional code will be discussed in the next section.

On a memoryless channel, there are three different procedures for decoding convolutional codes, viz. threshold decoding, sequential decoding, and Viterbi (maximum-likelihood) decoding. Sequential decoding was first introduced in 1957 by Wozencraft [2]. It can be applied to any convolutional code, however the complexity of its decoding computations is not fixed and can result in long delays in decoding. Threshold decoding was introduced by Massey [3] in 1963. It is a sub-optimum decoding procedure and its applicability is dependent on the individual code. In 1967, making use of the fact that an information digit can affect the coded digits in only a finite number of subsequent time periods, Viterbi proposed a new decoding algorithm [4] for decoding any convolutional code in the presence of noise due to a memoryless channel. His algorithm is a welcome alternative to the threshold decoding and sequential decoding algorithms. Unlike sequential decoding, the computational complexity for decoding a digit in the Viterbi algorithm is fixed. Furthermore it was proved later [5] that the Viterbi algorithm is actually a maximum likelihood decoding procedure. Practical decoders based on the Viterbi algorithm have actually been built and tested [6,7].

The Viterbi algorithm is a powerful procedure for decoding any convolutional code on a memoryless channel. For such channels the decoding procedure yields the minimum probability of error provided all input sequences are equally likely and is therefore optimum in this sense. Unfortunately, except for the space channel, most of the real channels on earth are not memoryless channels. They generally exhibit a certain degree of memory in their noise distributions and the errors tend to cluster together in bursts. Thus, for such real channels, it would be inappropraite to use the Viterbi algorithm which is designed for memoryless channels only. Using the algorithm in its present form would result in sub-optimum performance. In order to achieve optimum performance, the Viterbi maximum-likelihood decoding algorithm for the memoryless channel must be modified or generalized so that it will still be a maximum-likelihood decoding algorithm when used on such channels with memory.

2.2. PRELIMINARIES

The necessary basic understanding of convolutional codes will be presented in this section. For further detail refer to [8]. For ease of discussion, binary convolutional codes of rate 1/n will be considered here. Generalization to nonbinary codes and any other rate is straightforward. A rate 1/n convolutional code encoder is a linear finite-state machine consisting of a (k-1)-stage shift register and n modulo-2 adders which give the coded output, where K·n is the constraint length of the code. An example with K=3 and n=2 is shown in Fig. 1.

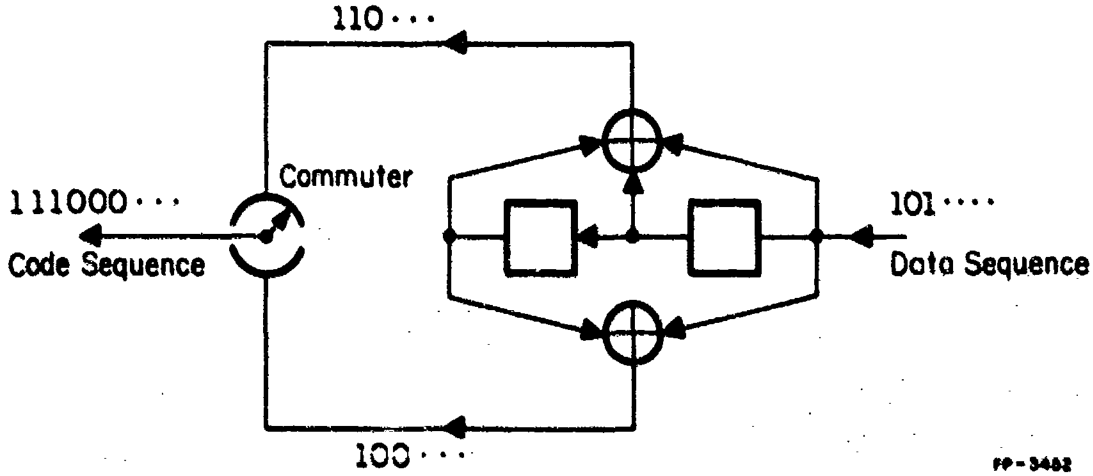

Fig. 1. A rate ½ encoder

During encoding the input data is shifted into the register one bit at a time, causing the encoder to instantaneously produce n encoded digits. This procedure continues until L data symbols are fed into the shift register. The result is a code with a tree structure having L branching levels. Each branch contains n encoded digits. An example of the tree structure with L=4 is given in Fig. 2 for the encoder of Fig. 1. Branching upwards corresponds to an input of 0 while branching downwards

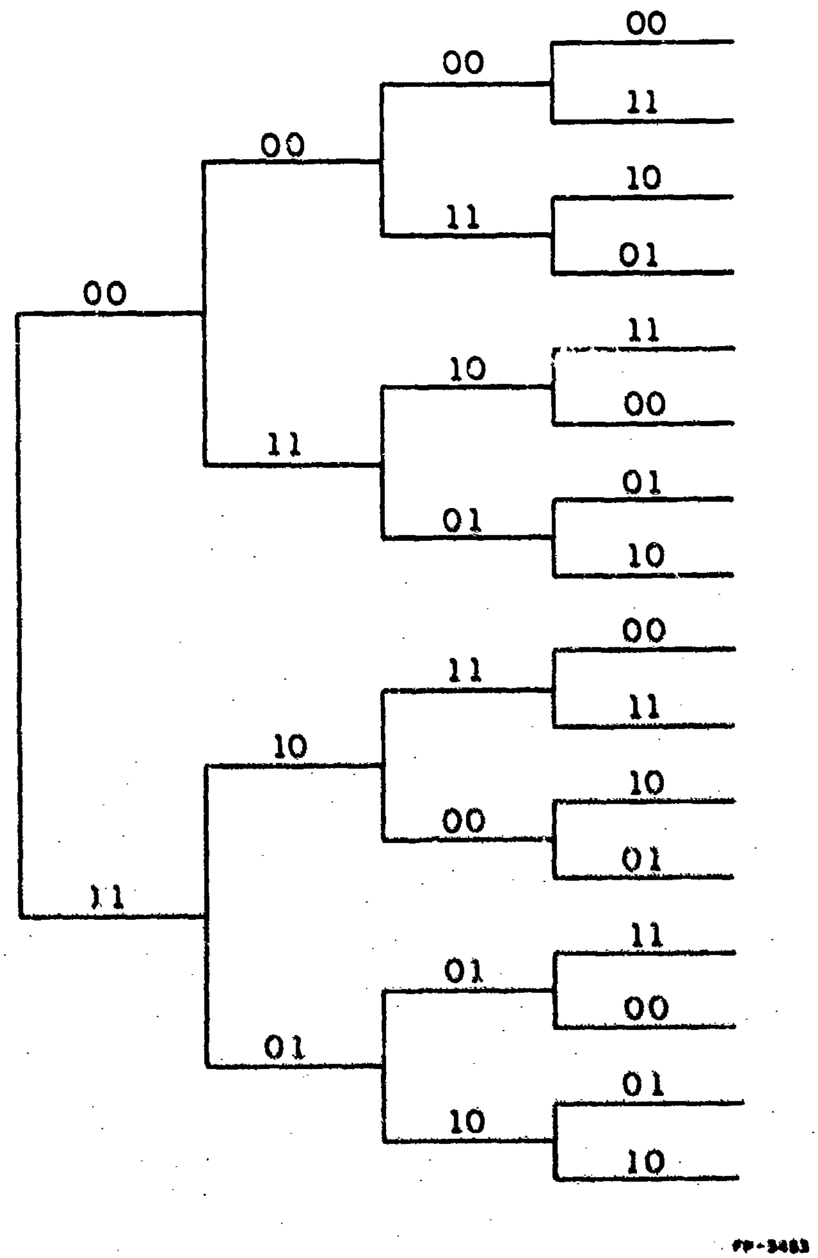

Fig. 2. Tree structure for encoder of Fig. 1 with L = 4

corresponds to an input of 1.

From Fig. 1 we can observe an important property of convolutional codes; that an input digit can affect the output digits in at most K time periods. This can also be seen from the fact that the convolutional encoder is a finite-state machine and the code can hence be represented by a state-diagram of 2^{K-1} states, the states being the contents of the (K-1)-stage shift register. The state diagram for the encoder of Fig. 1 is given in Fig. 3.

2.3. GENERALIZED VITERBI DECODING ALGORITHM

In this section the Viterbi decoding algorithm for the memoryless channel will be "re-invented" using an approach different from that used by Viterbi. Viterbi derived his algorithm [4] in an intuitive manner by observing the structure of the convolutional codes, and then proved that his algorithm was actually a maximum-likelihood decoding algorithm [5]. Here we shall start with the mathematical expression for maximum-likelihood decoding and then derive from it the mathematical formulation of the Viterbi algorithm. Since we start with the maximum-likelihood decoding and arrive at the Viterbi decoding algorithm, it is clear that the Viterbi decoding algorithm is a maximum-likelihood decoding algorithm. Using precisely the same approach we will then derive a generalized formulation of the Viterbi algorithm so that it can handle channels with finite memory. A channel is said to be of finite memory if its probability of a bit in error is dependent on a finite number of previous bits. In particular a channel is said to have memory m if

$$\Pr(e_i|e_{i-1}e_{i-2}\ldots) = \Pr(e_i|e_{i-1}e_{i-2}\ldots e_{i-m}) \qquad (1)$$

Let $a_1a_2\ldots a_L$ be the input data sequence to a (K-1)-stage shift register encoder with n outputs. The encoded sequence will then be a string of Ln symbols. Let $X_i = (x_{i1}x_{i2}\ldots x_{in})$ be the n-symbol output of the encoder when a_i is fed into the encoder, and let $Y_i = (y_{i1}y_{i2}\ldots y_{in})$ be the corresponding received n-symbol codeword. The errors added by the channel form $E = (e_{i1}e_{i2}\ldots e_{in}) = (y_{i1} + x_{i1}, y_{i2} + x_{i2}, \ldots y_{in} + x_{in})$. Because of the nature of the encoder each a_i can effect only $X_iX_{i+1}\ldots X_{i+K-1}$.

Given a received sequence $Y_1Y_2\ldots Y_L$, maximum-likelihood decoding would have to determine the data sequence $a_1a_2\ldots a_L$ for which the likelihood function $P(Y_1Y_2\ldots Y_L|a_1a_2\ldots a_L)$ is the greatest among all possible data sequences. In other words, the following operation must be performed:

$$\max_{a_1\ldots a_L} P(Y_1\ldots Y_L|a_1\ldots a_L) \qquad (2)$$

Since knowing $a_1\ldots a_L$ is equivalent to knowing $X_1\ldots X_L$, (2) is equivalent to

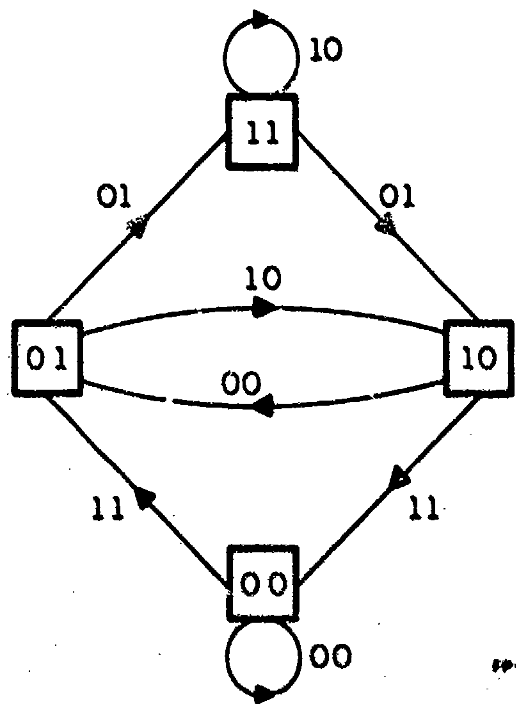

Fig. 3. State-diagram for encoder of Fig. 1

$$\underset{a_1 \cdots a_L}{\mathrm{Max}} \; P(E_1 \ldots E_L) \tag{3}$$

where $E_i = Y_i + X_i$, i.e., determining the most likely error pattern for the channel being used. Using the identity

$$P(AB) = P(A)P(B|A) \tag{4}$$

(2) can be written as

$$\underset{a_1 \cdots a_L}{\mathrm{Max}} \; P(Y_1 \ldots Y_K|a_1 \ldots a_L)P(Y_{K+1}|a_1 \ldots a_L, Y_1 \ldots Y_K)P(Y_{K+2}|a_1 \ldots a_L, Y_1 \ldots Y_{K+1}) \ldots$$
$$P(Y_{L-1}|a_1 \ldots a_L, Y_1 \ldots Y_{L-2})P(Y_L|a_1 \ldots a_L, Y_1 \ldots Y_{L-1}) \tag{5}$$

Let us first consider a memoryless channel. For such a channel the probability of receiving Y_i depends only on the X_i that was sent, which in turn is a function of only $a_{i-k+1} \ldots a_i$, i.e.,

$$P(Y_i|a_1 \ldots a_L, Y_1 \ldots Y_{i-1}) = P(Y_i|a_{i-K+1} \ldots a_i) \tag{6}$$

Substituting (6) into (5) we can rewrite (5) as:

$$\underset{a_1 \cdots a_L}{\mathrm{Max}} \; P(Y_1 \ldots Y_K|a_1 \ldots a_K)P(Y_{K+1}|a_2 \ldots a_{K+1}) \ldots P(Y_{L-1}|a_{L-K} \ldots a_{L-1})$$
$$P(Y_L|a_{L-K+1} \ldots a_L) \tag{7}$$

In (7) we note the fact that the second and higher terms are independent of a_1, and the third and higher terms are independent of a_2, etc. We can regroup (7) into the form:

$$\underset{a_{L-K+1} \cdots a_L}{\mathrm{Max}} \; P(Y_L|a_{L-K+1} \ldots a_L) \{ \underset{a_{L-K}}{\mathrm{Max}} \; P(Y_{L-1}|a_{L-K} \ldots a_{L-1}) \{ \ldots$$
$$\ldots \{ \underset{a_2}{\mathrm{Max}} \; P(Y_{K+1}|a_2 \ldots a_{K+1}) \{ \underset{a_1}{\mathrm{Max}} \; P(Y_1 \ldots Y_K|a_1 \ldots a_K)\}\} \ldots \}\} \tag{8}$$

In (8) each maximization procedure is over a single variable except the first one, which is over K-1 variables $a_{L-K+1} \ldots a_L$. If during encoding we agree to add K-1 zeros to the end of the data sequence $a_1 \ldots a_L$, (K-1)n more digits will be transmitted and received, namely $X_{L+1} \ldots X_{L+K-1}$ and $Y_{L+1} \ldots Y_{L+K-1}$ respectively. This additional sequence does not contain any new information but will be seen shortly to be very helpful in simplifying the decoding procedure. The maximum-likelihood decoding procedure of (2) would now be

$$\max_{a_1 \ldots a_L} P(Y_1 Y_2 \ldots Y_{L+K-1} | a_1 \ldots a_L \underbrace{0 \ldots 0}_{K-1}) \tag{9}$$

The K-1 zeros in (9) may be omitted since they are known quantities. Following the same procedure as before, an expression similar to (8) can be derived from (9) except that Y_L is now replaced by $Y_L \ldots Y_{L+K-1}$, i.e.,

$$\max_{a_{L-K+1}} P(Y_L \ldots Y_{L+K-1} | a_{L-K+1} \ldots a_L)\{ \max_{a_{L-K}} P(Y_{L-1} | a_{L-K} \ldots a_{L-1})\{ \ldots$$

$$\ldots\{ \max_{a_2} P(Y_{K+1} | a_2 \ldots a_{K+1})\{ \max_{a_1} P(Y_1 \ldots Y_k | a_1 \ldots a_K)\}\} \ldots \}\} \tag{10}$$

Using (4) and (6) $P(Y_L \ldots Y_{L+K-1} | a_{L-K+1} \ldots a_L)$ can again be broken up into K terms:

$$P(Y_L \ldots Y_{L+K-1} | a_{L-K+1} \ldots a_L) = P(Y_L | a_{L-K+1} \ldots a_L) P(Y_{L+1} | a_{L-K+2} \ldots a_L) \ldots$$

$$\ldots P(Y_{L+K-2} | a_{L-1} a_L) P(Y_{L+K-1} | a_L) \tag{11}$$

Noting once again that the second and higher terms of (11) are independent of a_{L-K+1}, etc., the maximization process over $a_{L-K+1} \ldots a_L$ in (10) can once again be broken into K maximizations each over a single variable:

$$\max_{a_L} P(Y_{L+K-1} | a_L)\{ \max_{a_{L-1}} P(Y_{L+K-2} | a_{L-1} a_L)\{ \ldots$$

$$\ldots\{ \max_{a_{L-K+1}} P(Y_L | a_{L-K+1} \ldots a_L)\{ \max_{a_{L-K}} P(Y_{L-1} | a_{L-K} \ldots a_{L-1})\{ \ldots$$

$$\ldots\{ \max_{a_2} P(Y_{K+1} | a_2 \ldots a_{K+L})\{ \max_{a_1} P(Y_1 \ldots Y_K | a_1 \ldots a_K)\}\} \ldots \} \tag{12}$$

Equation 12 is the mathematical expression of a maximum-likelihood decoding procedure for the convolutional code over a memoryless channel. Not too surprisingly it is the same as the Viterbi decoding algorithm. The decoding procedure as represented by (12) may be interpreted in the following way:

Step 1: Compute the likelihood functions $P(Y_1 \ldots Y_K | a_1 \ldots a_K)$ for all 2^K possible paths $a_1 \ldots a_K$. For each of the 2^{K-1} paths $a_2 \ldots a_K$ choose that a_1' which gives the greatest likelihood function and call it the survivor $A_1(a_2 \ldots a_K) = a_1'$.

<u>Step 2</u>: Compute the likelihood functions $P(Y_{K+1}|a_2 \ldots a_{K+1})$ and multiply by their corresponding previous likelihood functions $P(Y_1 \ldots Y_K|A_1(a_2 \ldots a_K)a_2 \ldots a_K)$ to form the new likelihood function $P(Y_1 \ldots Y_{K+1}|A_1(a_2 \ldots a_K)a_2 \ldots a_{K+1})$ for all 2^K possible paths $a_2 \ldots a_{K+1}$. For each path $a_3 \ldots a_{K+1}$ choose that a_2' which gives the greatest likelihood function and call the sequence $A_1(a_2'a_3 \ldots a_K)a_2'$ the survivor $A_2(a_3 \ldots a_{K+1})$.

<u>Step 3 -- Step L-K+1</u>: Proceed in a similar manner as in Step 2. In particular at the i-th step, $3 \leq i \leq L-K+1$, compute the 2^K likelihood functions $P(Y_{i+K-1}|a_i \ldots a_{i+K-1})$ and multiply by their corresponding previous likelihood functions $P(Y_1 \ldots Y_{i+K-2}|A_{i-1}(a_i \ldots a_{i+K-2})a_i \ldots a_{i+K-2})$ to form the new likelihood function $P(Y_1 \ldots Y_{i+K-1}|A_{i-1}(a_i \ldots a_{i+K-2})\ a_i \ldots a_{i+K-1})$ for all possible paths $a_i \ldots a_{i+K-1}$. For each path $a_{i+1} \ldots a_{i+K-1}$ choose that a_i' which gives the greatest likelihood function and call the sequence $A_{i-1}(a_i'a_{i+1} \ldots a_{i+K-2})a_i$ the survivor $A_i(a_{i+1} \ldots a_{i+K-1})$.

Step L-K+2 -- Step L-1: Proceed in a similar manner as before, except that the length of the path is shortened by one at the end of each step. In particular at the i-th step, $L-K+2 \leq i \leq L-1$, compute the 2^{L+1-i} likelihood functions for all possible paths $a_i \ldots a_L$. For each path $a_{i+1} \ldots a_L$ choose that a_i' which gives the greatest likelihood function, and call the sequence $A_{i-1}(a_i' \ldots a_L)a_i'$ the survivor $A_i(a_{i+1} \ldots a_L)$.

<u>Step L</u>: Compute the 2 likelihood functions for $a_L = 0$ and $a_L = 1$. Choose that a_L' which gives the greater likelihood function. The survivor sequence $A_{L-1}(a_L')a_L'$ is the maximum-likelihood decoded sequence.

From the above procedure it can be seen that at each step of the decoding, except the final K-1 steps, maximization has to be done for 2^{K-1} different paths. Since there are also 2^{K-1} states in the state-diagram of the code, the state-diagram can be used as a system diagram for the decoding algorithm. At the end of each decoding step each state (path) remembers its survivor and corresponding likelihood function. During the next decoding period all the possible state transitions are made and the corresponding new likelihood functions computed. Then at each state the new likelihood functions are compared and the new survivor is chosen, and the system is ready for another decoding period. During the final K-1 steps the same thing happens, only that now a decreasing number of states would be involved and an increasing number of states would become idle.

We have seen so far that the Viterbi decoding algorithm for the memoryless channel can be directly derived from the general maximum-likelihood decoding formulation (2) by making use of properties (6) of the memoryless channel. Using precisely the same approach, we shall now show that the Viterbi decoding algorithm can easily be generalized to handle channels with finite memory.

Let us consider a channel with finite memory m. Let J be the smallest integer greater than or equal to m/n. For such a channel, the probability of receiving the initial sequence $Y_1 \ldots Y_J$ will not only depend on $a_1 \ldots a_J$, but also on the error state $e_{-m+1} \ldots e_{-1} e_0$ of the channel before the first digit y_{11} is received. If we agree to transmit m zeros just before we transmit the first coded digit x_{11}, the received digits corresponding to these m zeros would tell us the error state $e_{-m+1} \ldots e_0$ of the channel. Then the maximum-likelihood decoding formulation of (2) can be slightly modified to

$$\underset{a_1 \ldots a_L}{\text{Max}} \; P(Y_1 \ldots Y_L | a_1 \ldots a_L, e_{-m+1} \ldots e_0) \tag{15}$$

Furthermore if during encoding, we agree to add K+J-1 zeros to the end of the data sequence $a_1 \ldots a_L$, (K+J-1)n more digits will be transmitted and received, namely $X_{L+1} \ldots X_{L+K+J-1}$ and $Y_{L+1} \ldots Y_{L+K+J-1}$ respectively. Just as in the memoryless channel case, this additional sequence does not contain any new information but will also be seen to simplify the decoding procedure. The maximum-likelihood decoding procedure of (15) would now be

$$\underset{a_1 \ldots a_L}{\text{Max}} \; P(Y_1 \ldots Y_{L+K+J-1} | a_1 \ldots a_L, e_{-m+1} \ldots e_0). \tag{16}$$

Applying identity (4) to (16) as before, (16) can be written as

$$\begin{aligned} \underset{a_1 \ldots a_L}{\text{Max}} \; & P(Y_1 \ldots Y_{K+J} | a_1 \ldots a_L, e_{-m+1} \ldots e_0) \, P(Y_{K+J+1} | a_1 \ldots a_L, Y_1 \ldots Y_{K+J}, \\ & e_{-m+1} \ldots e_0) \ldots P(Y_{L+K+J-2} | a_1 \ldots a_L, Y_1 \ldots Y_{L+K+J-3}, e_{-m+1} \ldots e_0) \\ & P(Y_{L+K+J-1} | a_1 \ldots a_L, Y_1 \ldots Y_{L+K+J-2}, e_{-m+1} \ldots e_0) \end{aligned} \tag{17}$$

For the channel with memory m, the probability of receiving Y_i, $i > J$, will depend not only on X_i, which gives the error pattern $E_i = Y_i + X_i$, but also on the previous error sequence $E_{i-J} \ldots E_{i-1} = (Y_{i-J} \ldots Y_{i-1}) + (X_{i-J} \ldots X_{i-1})$. The sequence $X_{i-J} \ldots X_i$ as a whole is a function of $a_{i-J-K+1} \ldots a_i$. Thus for this channel

$$P(Y_i|a_1\ldots a_L, Y_1\ldots Y_{i-1}, e_{-m+1}\ldots e_0) = P(Y_i|a_{i-J-K+1}\ldots a_i, Y_{i-J}\ldots Y_{i-1}) \quad (18)$$

The likelihood function on the right-hand side of (18) is easy to compute since

$$\begin{aligned} & P(Y_i|a_{i-J-K+1}\ldots a_i, Y_{i-J}\ldots Y_{i-1}) \\ &= P(Y_i|X_{i-J}\ldots X_i, Y_{i-J}\ldots Y_{i-1}) \\ &= P(Y_i + X_i|(Y_{i-J}\ldots Y_{i-1}) + (X_{i-J}\ldots X_{i-1})) \\ &= P(E_i|E_{i-J}\ldots E_{i-1}). \end{aligned} \quad (19)$$

which can be computed from the channel model parameters. Substituting (18) into (17) we can rewrite (17) as

$$\begin{aligned} \underset{a_1\ldots a_L}{\text{Max}} \; & P(Y_1\ldots Y_{K+J}|a_1\ldots a_{K+J}, e_{-m+1}\ldots e_0)P(Y_{K+J+1}|a_2\ldots a_{K+J+1}, Y_{K+1}\ldots Y_{K+J}) \\ & \ldots P(Y_{L+K+J-2}|a_{L-1}a_L, Y_{L+K-1}\ldots Y_{L+K+J-3})P(Y_{L+K+J-1}|a_L, \\ & \qquad Y_{L+K-1}\ldots Y_{L+K+J-2} \end{aligned} \quad (20)$$

Once again we note the fact that the second and higher terms of (20) are independent of a_1 and the third and higher terms are independent of a_2, etc., we can regroup (20) into the form:

$$\begin{aligned} \underset{a_L}{\text{Max}} \; & P(Y_{L+K+J-1}|a_L, Y_{L+K-1}\ldots Y_{L+K+J-2})\{\underset{a_{L-1}}{\text{Max}} \; P(Y_{L+K+J-2}|a_{L-1}a_L, \\ & Y_{L+K-2}\ldots Y_{L+K+J-3})\{ \ldots\{\underset{a_2}{\text{Max}} \; P(Y_{K+J+1}|a_2\ldots a_{K+J+1}, Y_{K+1}\ldots Y_{K+J})\{ \\ & \{\underset{a_1}{\text{Max}} \; P(Y_1\ldots Y_{K+J}|a_1\ldots a_{K+J}, e_{-m+1}\ldots e_0)\}\}\ldots\}\} \end{aligned} \quad (21)$$

Equation (21) is the mathematical expression of a maximum-likelihood decoding procedure for the convolutional code over a channel with finite memory m. We shall call it the generalized Viterbi decoding algorithm. The decoding procedure as represented by (21) may be interpreted in the following way:

<u>Step 1</u>: Compute the likelihood functions $P(Y_1\ldots Y_{K+J}|a_1\ldots a_{K+J}, e_{-m+1}\ldots e_0)$ for all 2^{K+J} possible paths $a_1\ldots a_{K+J}$. For each of

the 2^{K+J-1} paths $a_2 \ldots a_{K+J}$ choose that a_1' which gives the greatest likelihood function and call it the survivor $A_1(a_2 \ldots a_{K+J}) = a_1'$.

Step 2: Compute the likelihood function $P(Y_{K+J+1} | a_2 \ldots a_{K+J+1}, Y_{K+1} \ldots Y_{K+J})$ and multiply by their corresponding previous likelihood functions $P(Y_1 \ldots Y_{K+J} | A_1(a_2 \ldots a_{K+J}) a_2 \ldots a_{K+J}, e_{-m+1} \ldots e_0)$ to form the new likelihood function $P(Y_1 \ldots Y_{K+J+1} | A_1(a_2 \ldots a_{K+J}) a_2 \ldots a_{K+J+1}, e_{-m+1} \ldots e_0)$ for all 2^{K+J} possible paths $a_2 \ldots a_{K+J+1}$. For each path $a_3 \ldots a_{K+J+1}$ choose that a_2' which gives the greatest likelihood function and call the sequence $A_1(a_2' a_3 \ldots a_{K+J}) a_2'$ the survivor $A_2(a_3 \ldots a_{K+J+1})$.

Step 3 -- Step L-K-J+1: Proceed in a similar manner as in Step 2. In particular at the i-th step, $3 \leq i \leq L-K-J+1$, compute the 2^K likelihood functions $P(Y_{i+K+J-1} | a_i \ldots a_{i+K+J-1}, Y_{i+K-1} \ldots Y_{i+K+J-2})$ and multiply by the corresponding previous likelihood functions $P(Y_1 \ldots Y_{i+K+J-2} | A_{i-1}(a_i \ldots a_{i+K+J-2}) a_i \ldots a_{i+K+J-2}, e_{-m-i} \ldots e_0)$ to form the new likelihood function $P(Y_1 \ldots Y_{i+K+J-1} | A_{i-1}(a_i \ldots a_{i+K+J-2}) a_i \ldots a_{i+K+J-1}, e_{-m+1} \ldots e_0)$ for all possible paths $a_i \ldots a_{i+K+J-1}$. For each path $a_{i+1} \ldots a_{i+K+J-1}$ choose that a_i' which gives the greatest likelihood function and call the sequence $A_{i-1}(a_i' a_{i+1} \ldots a_{i+K+J-2}) a_i'$ the survivor $A_i(a_{i+1} \ldots a_{i+K+J-1})$.

Step L-K-J+2 -- Step L-1: Proceed in a similar manner as before, except that the length of the path is shortened by one at the end of each step. In particular, at the i-th step, $L-K-J+2 \leq i \leq L-1$, compute the 2^{L+1-i} likelihood functions for all possible paths $a_i \ldots a_L$. For each path $a_{i+1} \ldots a_L$ choose that a_i' which gives the greatest likelihood function, and call the sequence $A_{i-1}(a_i' \ldots a_L) a_i'$ the survivor $A_i(a_{i+1} \ldots a_L)$.

Step L: Compute the 2 likelihood function for $a_L = 0$ and $a_L = 1$. Choose that a_L' which gives the greatest likelihood function. The survivor sequence $A_{L-1}(a_L') a_L'$ is the maximum-likelihood decoded sequence.

From the above it can easily be seen that the decoding procedure as represented (21) is indeed a generalized Viterbi decoding algorithm since it contains the Viterbi algorithm as a special case when the memory of the channel m is equal to zero. For the channel with memory, it is seen that at each step of the decoding, except the final K+J-1 steps, maximization has to be done for

2^{K+J-1} different paths. Since there are only 2^{K-1} states in the state-diagram of the code, the state-diagram cannot be used as a system diagram for the decoding algorithm as in the memoryless channel case. However, for any 2^{K-1}-states state-diagram it is possible for one to expand it into a 2^{K+J-1}-states state-diagram. The easiest way to see this is to look at the encoder. One can add J stages of dummy shift register to the original K-1 stages of shift register in the encoder and then consider it as a K+J-1 states finite-state machine. This "new" encoder can thus now be represented by a state-diagram with 2^{K+J-1} states. An example of such a procedure for the encoder of Fig. 1 with J = 1 is as shown in Fig. 4. The expanded state-diagram obtained by the procedure just described may now be used as a system diagram for the generalized Viterbi algorithm in exactly the same manner as in the memoryless channel case. Thus the complexity of a generalized Viterbi decoder with parameters K and J is about the same as that of a Viterbi decoder for the memoryless channel with parameters K' = K+J.

2.4 EXAMPLE

As a very primitive example of comparing the performance of the generalized Viterbi decoder with that of a Viterbi decoder for the memoryless channel of the same complexity, let us use the code generated by the K = 3 encoder of Fig. 1 for the memoryless Viterbi decoder and use the code generated by the K = 2 encoder of Fig. 5 for the generalized Viterbi decoder. Assume L = 3 in both cases. Furthermore let us choose a very simple channel model representing a channel with finite memory m = 2. Such a model is completely specified by the following set of conditional probabilities:

$$\begin{aligned} P(1|00) &= 10^{-3} \\ P(1|01) &= 0.5 \\ P(1|10) &= 0.5 \\ P(1|11) &= 0.5 \end{aligned} \tag{22}$$

where a 1 indicates a channel error and a 0 indicates no error. Since m = n = 2, J = 1 for the generalized Viterbi decoder and the two decoders have the same degree of complexity with regard to hardware configuration and number of states.

Let us assume that the channel is error-free when the actual transmission begins, i.e., $e_0 = e_{-1} = 0$. Since both of the codes are linear codes, we can form a standard array for each of the codes. For the K = 3 code, choose the vector with the minimum weight in each coset as the coset leader. These would be the correctable error patterns chosen by the memoryless Viterbi decoder. For the K = 2 code choose the vector with the highest probability, given $e_0 = e_{-1} = 0$, in each coset as its coset leader. These would be the correctable error patterns chosen by the generalized Viterbi decoder.

If no coding is used at all in the channel, the probability of error P_{E_n} is

$$\begin{aligned} P_{E_n} &= 1 - P(00\ldots0|00) \\ &= 9.955 \times 10^{-3} \end{aligned}$$

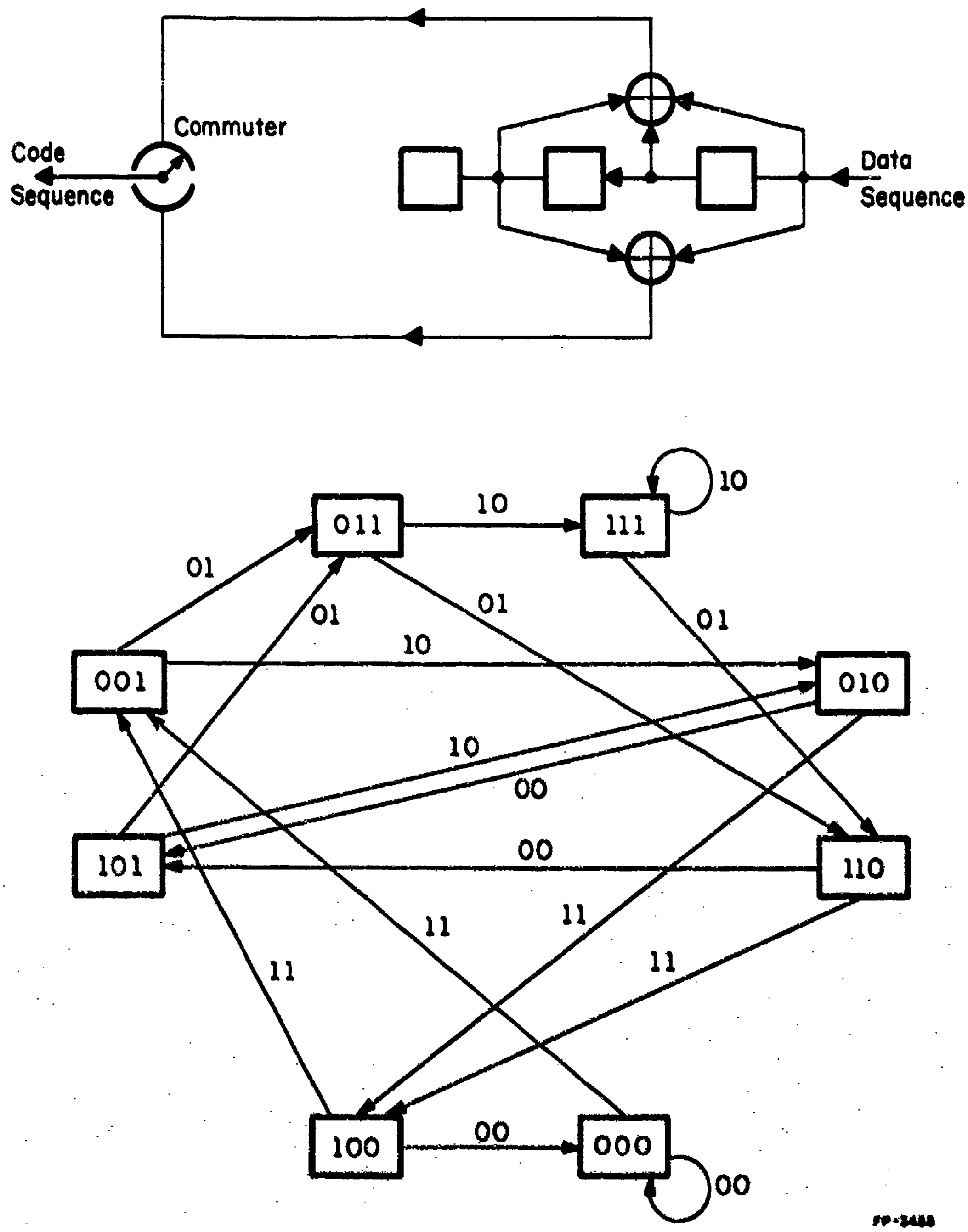

Fig. 4. Expanded encoder and state-diagram for encoder of Fig. 1 with J=1.

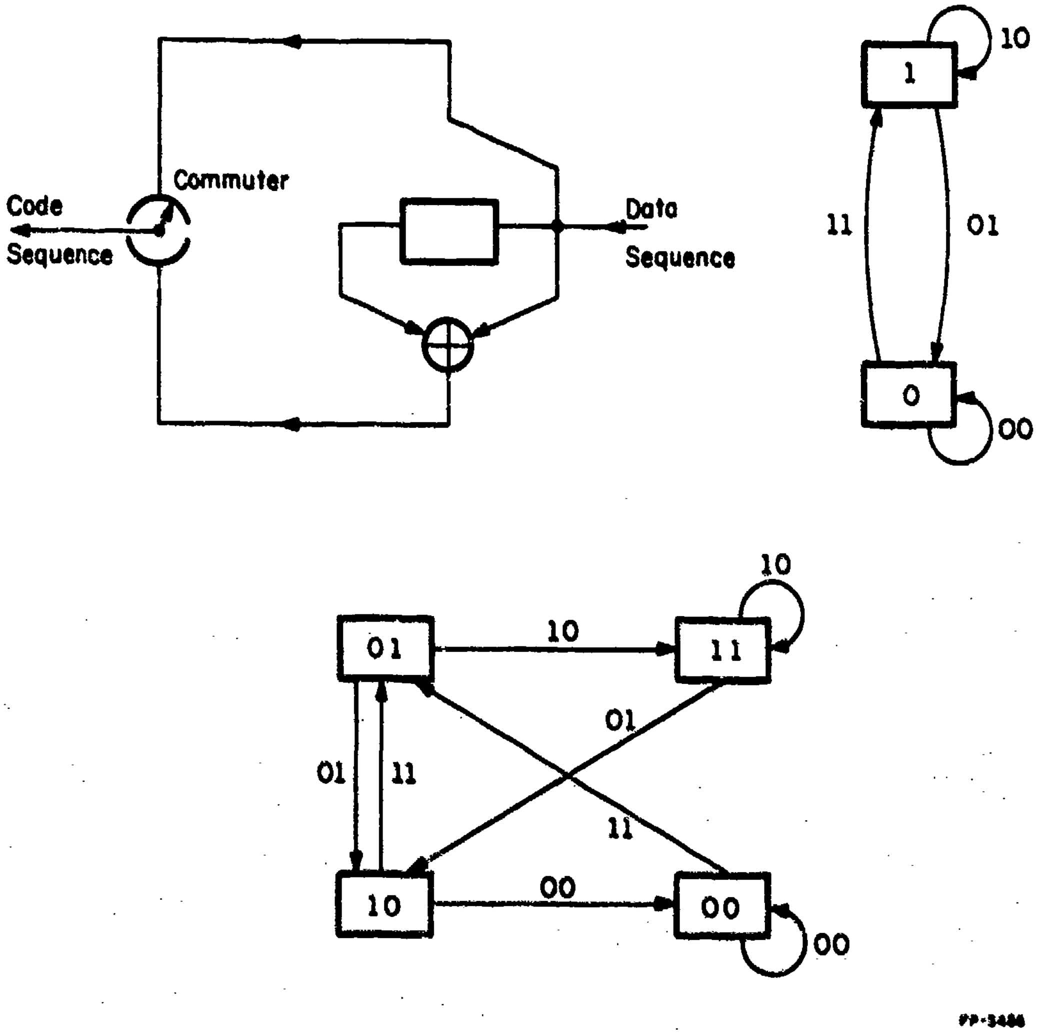

Fig. 5. Encoder, state-diagram, and expanded state-diagram (J-1) of a K=2 code

If the K = 3 code is used in conjunction with the memoryless Viterbi decoder, the probability of error P_{E_m} is

$$P_{E_m} = 1 - P(0\ldots0|00) - \sum_{E\epsilon S} P(E|00)$$

$$= 3.549 \times 10^{-3}$$

where S is the set of all correctable error patterns. For the K = 2 code and the generalized Viterbi decoder, the probability of error $P_{E_g} = 2.583 \times 10^{-3}$.

Thus the generalized Viterbi decoder slightly outperforms the memoryless channel Viterbi decoder.

2.5. CONCLUDING REMARKS

In this report we have developed a generalized version of the Viterbi decoding algorithm. This generalized algorithm can be used to perform maximum-likelihood decoding on any channel with finite-memory and is thus an optimum decoding algorithm for such channels. It is pointed out that the complexity of a generalized Viterbi decoder with parameters K and J is about the same as that of a Viterbi decoder for the memoryless channel with parameter K' = K+J. In a simple example we have seen that the generalized Viterbi algorithm indeed outperforms the memoryless Viterbi algorithm when the complexities of the two decoders are about the same. Although the same result has not yet been proved to be true for all channels with finite memory and all possible codes, it shows at the very least, that in certain cases the generalized Viterbi decoding algorithm is superior to the memoryless Viterbi algorithm when the complexity of each decoder is kept the same. Furthermore, if it is the complexity of the encoder rather than the decoder that is kept the same, then the generalized Viterbi algorithm which performs maximum-likelihood decoding will definitely be superior. Thus, in those systems where the cost of the encoder is the main concern and the cost of the decoder is of little concern, the generalized Viterbi algorithm should definitely be used. Just one such example is when there is a large number of sources (thus encoders) transmitting data to a data processing center which uses a general purpose computer as decoder and thus the cost of implementing either decoding algorithm is the same.

REFERENCES FOR SECTION 2

[1] P. Elias, "Coding for Noisy Channels," 1955 IRE International Convention Record, pt. 4, pp. 37-46.

[2] J. M. Wozencraft, "Sequential Decoding for Reliable Communication," 1957 IRE National Convention Record, Vol. 5, pt. 2, pp. 11-25.

[3] J. L. Massey, Threshold Decoding, Cambridge, Mass., MIT Press, 1963.

[4] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymtotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, IT-13, April 1967, pp. 260-269.

[5] A. J. Viterbi and J. P. Odenwalder, "Further Results on Optimal Decoding of Convolutional Codes," IEEE Transactions on Information Theory (Corresp.), IT-15, November 1969, pp. 732-734.

[6] J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," IEEE Transactions on Communication Technology, COM-19, October 1971, pp. 835-848.

[7] A. R. Cohen, J. A. Heller, and A. J. Viterbi, "A New Coding Technique for Asynchronous Multiple Access Communication," IEEE Transactions on Communication Technology, COM-19, October 1971, pp. 849-855.

[8] G. D. Forney, Jr., "Convolutional Codes I: Algebraic Structure," IEEE Transactions on Information Theory, IT-16, November 1970, pp. 720-738.

SECTION 3

MARKOV CHAIN MODEL

3.1. A REVIEW OF THE M-STATE MARKOV MODEL

The M-state Markov model is described by the transition probability matrix $P = \{p_{ij}\}$ and a set of probabilities of error in each state $h'_j = 1 - h_j$. M x M parameters are required to determine the model completely. It has been contemplated that a set of the coefficients A_i and $A_i(n_j, n_{j+1})$ of the unconditional and conditional gap distributions can be used to identify these parameters. However, a close examination revealed that the process of identification is considerably more complicated than it has been anticipated.

The elements of the matrix D are defined [1] by

$$D_{ij} = p_{ij} h_j \tag{1}$$

D can be diagonalized as follows:

$$D = G \underline{\alpha} G^{-1} \tag{2}$$

where $\underline{\alpha}$ is a diagonal matrix whose elements α_i are the eigenvalues of the matrix D. G is a non-singular matrix whose columns are eignevectors corresponding to α_i. Since the eigenvectors are unique up to a scalar, they can be chosen such that the <u>sum of the columns of G</u> is equal to unity, i.e.

$$\sum_{i=1}^{M} G_{ij} = 1 \ , \ j = 1,2,\ldots,M. \tag{3}$$

It has been shown [2] that

$$P(m) = \sum_{i=1}^{M} A_i \alpha_i^{\,m} \tag{4}$$

$$P(m/n) = \sum_{i=1}^{M} A_i(n) \alpha_i^{\,m} \tag{5}$$

and

$$P(m/n_j \leq n \leq n_{j+1}) = \sum_{i=1}^{M} A_i(n_j, n_{j+1}) \alpha_i^{\,m} \tag{6}$$

It has also been shown [3] that

$$P(m) = \underline{x}^T D^m \underline{c} \tag{7}$$

$$P(m/n) = \frac{\underline{x}^T D^n D' D^m \underline{e}}{\underline{x}^T D^n \underline{e} - \underline{x}^T D^{n+1} \underline{e}} \tag{8}$$

and

$$P(m/n_j \leq n \leq n_{j+1}) = \frac{\sum_{n=n_j}^{n_{j+1}} \underline{x}^T D^n D' D^m \underline{e}}{\underline{x}^T D^{n_j} \underline{e} - \underline{x}^T D^{n_{j+1}+1} \underline{e}} \tag{9}$$

Consider the case of $M = 3$, by definition

$$P(m) = \underline{x}^T D^m \underline{e} = \underline{x}^T G \underline{\alpha}^m G^{-1} \underline{e}$$

$$= (x_1 x_2 x_3) \begin{pmatrix} G_{11} G_{12} G_{13} \\ G_{21} G_{22} G_{23} \\ G_{31} G_{32} G_{33} \end{pmatrix} \begin{pmatrix} \alpha_1^m & 0 & 0 \\ 0 & \alpha_2^m & 0 \\ 0 & 0 & \alpha_3^m \end{pmatrix} \begin{pmatrix} G_{11}^{(-1)} G_{12}^{(-1)} G_{13}^{(-1)} \\ G_{21}^{(-1)} G_{22}^{(-1)} G_{23}^{(-1)} \\ G_{31}^{(-1)} G_{32}^{(-1)} G_{33}^{(-1)} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{aligned} \underline{x}^T D^m \underline{e} = {} & (G_{11}^{(-1)} + G_{13}^{(-1)} + G_{13}^{(-1)})(x_1 G_{11} + x_2 G_{21} + x_3 G_{31})\, \alpha_1^m \\ & + (G_{21}^{(-1)} + G_{22}^{(-1)} + G_{23}^{(-1)})(x_1 G_{12} + x_2 G_{22} + x_3 G_{32})\, \alpha_2^m \\ & + (G_{31}^{(-1)} + G_{32}^{(-1)} + G_{33}^{(-1)})(x_1 G_{13} + x_2 G_{23} + x_3 G_{33})\, \alpha_3^m \end{aligned} \tag{10}$$

now

$$x_i = \frac{p_i h_i'}{\sum_{i=1}^{3} p_i h_i'} = \frac{p_i h_i'}{p_e} \tag{11}$$

where p_i is the stationary probability of i-th state. The p_i satisfy the

following relations:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} p_{11} p_{12} p_{13} \\ p_{21} p_{22} p_{23} \\ p_{31} p_{32} p_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \tag{12}$$

and

$$p_1 + p_2 + p_3 = 1 \tag{13}$$

The stationary probabilities p_i can be expressed in terms of the transition probabilities p_{ij} as follows:

$$p_1 = \frac{(1-p_{22})(1-p_{33}) + p_{23}p_{32}}{(1-p_{22})(1-p_{33}) + p_{23}p_{32} + p_{21}(1-p_{33}) + p_{23}p_{31} + p_{31}(1-p_{22}) + p_{32}p_{21}} \tag{14}$$

$$p_2 = \frac{(1-p_{11})(1-p_{33}) + p_{13}p_{31}}{(1-p_{11})(1-p_{33}) + p_{13}p_{31} + p_{12}(1-p_{33}) + p_{13}p_{32} + p_{32}(1-p_{11}) + p_{31}p_{12}} \tag{15}$$

$$p_3 = \frac{(1-p_{11})(1-p_{22}) + p_{12}p_{21}}{(1-p_{11})(1-p_{22}) + p_{12}p_{21} + p_{13}(1-p_{22}) + p_{12}p_{23} + p_{23}(1-p_{11}) + p_{21}p_{13}} \tag{16}$$

The probability transition matrix P can be expressed in terms of D and in turn, expressed in terms of G and G^{-1}.

$$P = D \frac{1}{\underline{h}} = G \,\underline{\alpha}\, G^{-1} \frac{1}{\underline{h}}$$

or explicitly

$$\begin{pmatrix} p_{11} p_{12} p_{13} \\ p_{21} p_{22} p_{23} \\ p_{31} p_{32} p_{33} \end{pmatrix} = \begin{pmatrix} G_{11} G_{12} G_{13} \\ G_{21} G_{22} G_{23} \\ G_{31} G_{32} G_{33} \end{pmatrix} \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix} \begin{pmatrix} G_{11}^{(-1)} G_{12}^{(-1)} G_{13}^{(-1)} \\ G_{21}^{(-1)} G_{22}^{(-1)} G_{23}^{(-1)} \\ G_{31}^{(-1)} G_{32}^{(-1)} G_{33}^{(-1)} \end{pmatrix} \begin{pmatrix} \frac{1}{h_1} & 0 & 0 \\ 0 & \frac{1}{h_2} & 0 \\ 0 & 0 & \frac{1}{h_3} \end{pmatrix}$$

Carrying out the matrix multiplication, the result is

$$p_{ij} = \frac{1}{h_j} (\alpha_1 G_{i1} G_{1j}^{(-1)} + \alpha_2 G_{i2} G_{2j}^{(-1)} + \alpha_3 G_{i3} G_{3j}^{(-1)})$$

$$= \frac{1}{h_j} \sum_{k=1}^{3} \alpha_k G_{ik} G_{kj}^{(-1)} \tag{17}$$

Substituting (17) into (14), (15), and (16), into (11); and into (10), it is not difficult to realize that the magnitude of complexity is enormous.

3.2. AN ALTERNATIVE APPROACH

The complexity in identifying the parameters of the general Markov model stems from the fact that there are too many parameters to be determined. Some of the parameters could be specified beforehand as 0 or 1 to reduce the complexity. Several simpler models were discussed in both Quarterly Reports No. 2 and No. 4. Among them the simple partitioned Markov chain model has been extensively studied [4],[5],[6]. This model is completely determined by 2(M-1) parameters instead of M^2 parameters for the general Markov model, and these 2(M-1) parameters can be uniquely derived from the unconditional gap distribution which is, sometime, referred to as error free run distribution.

The general Markov model, were it possible to be derived from the unconditional and conditional gap distributions, would yield the same unconditional gap distribution as the simple partitioned Markov chain model. It would also yield the conditional gap distributions while the simple partitioned Markov model will not exhibit the interdependence between the gaps because it has only a single error state.

Investigation of the effect of these second order statistics, namely, the interdependence on the error burst distribution is under way. A computer program has been written to generate error sequences from the probability transition matrix characterizing the simple partitioned Markov model. Burst distributions are calculated from three sources: the original error sequence, the error sequence generated from the gap model and the error sequence generated from the simple Markov model. Some preliminary results have been obtained and are presently under study.

3.3. ERROR BURST

In Quarterly Report No. 4, the error burst is defined [7] as a sequence of bits starting and ending with an error and separated from neighboring bursts by at least K error free bits, where K is a parameter. A second definition [6] is instrumental in evaluating some error correcting codes. It defines the error burst with error density Δo as follows:

(1) A burst begins with an error and ends with an error;

(2) The ratio of the number of errors to the total number of bits of a burst is larger than or equal to the specified density Δo;

(3) If successive inclusion of the next error keeps the error density above Δo, the burst continues; otherwise the burst ends;

(4) A burst cannot begin with an error belonging to the previous burst;

(5) A single error is defined to be a single-error burst with burst length of one digit.

A computer program has been written to compute the burst distribution from the error sequence using the second definition. Some preliminary result on the burst distribution from the original error data is shown in the following figure.

The two different definitions do not cause appreciable change in the burst statistics. This can be seen by the following reasoning. The first definition does not allow a long string of "o" in the burst while the second definition does. However, the probability of these bursts is small because a long string of "o" inside the burst must be preceded by dense errors.

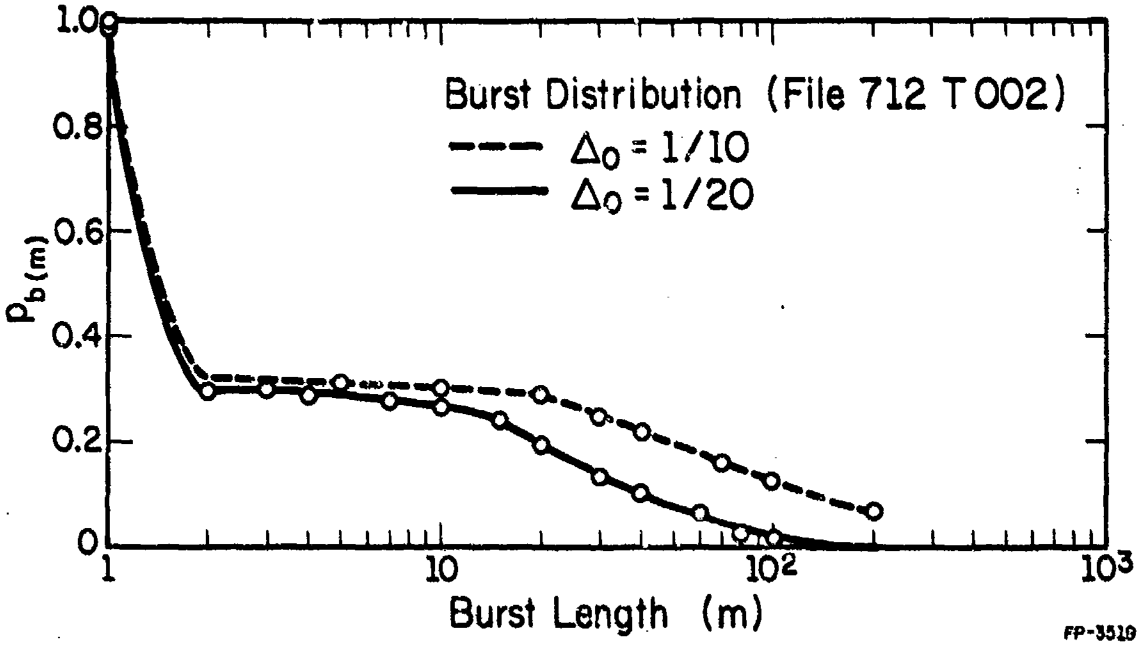

Fig. 6. Burst distribution

REFERENCES FOR SECTION 3

[1] R. T. Chien, F. P. Preparata, A. H. Haddad and C. L. Chen, Third Quarterly Progress Report, Section 4, Contract No. DAAB-07-71-C-0292 for U. S. Army Electronics Command, Fort Monmouth, New Jersey, June 1972.

[2] R. T. Chien, F. P. Preparata, A. H. Haddad, and C. L. Chen, Second Quarterly Progress Report, Section 2.2, Contract No. DAAB-07-71-C-0292 for U. S. Army Electronics Command, Fort Monmouth, New Jersey, January 1972.

[3] Detailed Derivation Eq. 31 and Eq. 32.

[4] B. D. Fritchman, "A Binary Channel Characterization Using Partitioned Markov Chains," IEEE Trans. Information Theory, Vol. IT-13, pp. 221-227, April 1967.

[5] S. Tsai, "Markov Characterization of the HF Channel," IEEE Trans. on Communication Technology, Vol. COM-17, pp. 24-32, February 1969.

[6] S. Tsai, "Analytic Method of Evaluating Error Correcting Codes," IEEE International Conference on Communications, Conference Proceedings, 1972.

[7] R. T. Chien, F. P. Preparata, A. H. Haddad and C. L. Chen, Fourth Quarterly Progress Report, Section 2, Contract No. DAAB-07-71-C-0292 for U. S. Army Electronics Command, Fort Monmouth, New Jersey, October 1972.

Distribution List

101 Defense Documentation Center
ATTN: DDC-TCA
Cameron Station (Bldg 5)
12 Alexandria, Virginia 22314

104 Defense Communications Agency
ATTN: Code 340
1 Washington, D. C. 20305

107 Director
National Security Agency
Attn: TDL
1 Fort George G. Meade, Maryland 20755

110 DCA System Engineering Facility
ATTN: Code T202; Technical Library
1860 Wiehle Avenue
1 Reston, Virginia 22090

200 Office of Naval Research
Code 427
1 Arlington, Va. 22217

205 Director
Naval Research Laboratory
ATTN: Code 2627
1 Washington, D. C. 20390

206 Commander
Naval Electronics Laboratory Center
ATTN: Library
1 San Diego, California 92152

210 Commandant, Marine Corps
Hq, U. S. Marine Corps
ATTN: Code A04C
1 Washington, D. C. 20380

301 Rome Air Development Center (TDLD)
ATTN: Documents Library
1 Griffiss AFB, New York 13440

307 HQ ESD(TRI)
L. G. Hanscom Field
1 Bedford, Massachusetts 01730

403 HQDA(DACE-ED)
1 Washington, D. C. 20314

405 OFC, Asst. Sec. of the Army
(R&D)
ATTN: Asst. for Research
Room 3-E-379, The Pentagon
1 Washington, D. C. 20310

409 Commanding General
U. S. Army Materiel Command
ATTN: AMCMA-EE
1 Washington, D. C. 20315

415 Commanding General
U. S. Army Materiel Command
ATTN: AMCRD-O
1 Washington, D. C. 20315

421 CG, U.S. Army Missile Command
Redstone Scientific Info Center
ATTN: Chief, Document Section
1 Redstone Arsenal, Alabama 35809

428 Commanding General
U.S. Army Combat Developments
Command
ATTN: CDCMS-E
1 Fort Belvoir, Va. 22060

480 Commanding Officer
USASA Test & Evaluation Center
1 Fort Huachuca, Arizona 85613

483 Commander
U.S. Army Research Ofc.
(Durham)
Box CM-Duke Station
1 Durham, North Carolina 27706

Defense Communication Agency
ATTN: Code T222,
Dr. R. Crawford
1 Washington, D. C. 20305

504 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-R (H. Cohen)
1 Washington, D. C. 20315

596 Commanding Officer
U.S. Army Combat Developments Command
Communications-Electronics Agency
1 Fort Monmouth, New Jersey 07703

599 TRI-TAC Office
ATTN: CSS(Dr. Pritchard)
1 Fort Monmouth, New Jersey 07703

605 U. S. Army Liaison Office
MIT-Lincoln Laboratory, Room A-210
P.O. Box 73
1 Lexington, Massachusetts 02173

U.S. Navy Electronics Center
ATTN: Mr. Ray Kelly
Code 3200
1 San Diego, California 92152

Rome Air Development Center
ATTN: CORS, Mr. Miles Bicklehaupt
1 Griffiss Air Force Base, New York 13440

680 Commanding General
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

2 AMSEL-TD-TI

1 AMSEL-RD-MT

15 AMSEL-NL-R-2

1 AMSEL-NL-R-5

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
Analytic Mathematical Models of Tactical Military Communications Channels

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Quarterly Report 1 October 1972 - 31 December 1972

5. AUTHOR(S) (First name, middle initial, last name)
Professors R. T. Chien, C. L. Chen and S. Tsai

6. REPORT DATE
May 1973

7a. TOTAL NO. OF PAGES
27

7b. NO. OF REFS
15

8a. CONTRACT OR GRANT NO.
DAAB07-71-C-0292

b. PROJECT NO. 1S6.62701.A327

c. 06

d. 071

9a. ORIGINATOR'S REPORT NUMBER(S)
ECOM-0292-6

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Distribution limited to US Government Agencies only; Test and Evaluation; March 1973. Other request for this document must be referred to Commander, US Army Electronics Command, ATTN: AMSEL-NL-R-2, Fort Monmouth, New Jersey 07703

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
US Army Electronics Command
ATTN: AMSEL-NL-R-2
Fort Monmouth, New Jersey 07703

13. ABSTRACT

The Viterbi decoding algorithm yields minimum probability of error when applied to a memoryless channel provided that all input sequences are equally likely. In this report, the algorithm was generalized for application to channels with finite memory and it was shown that the generalized algorithm is also maximum likelihood decoding. It was also shown that the generalized Viterbi algorithm on a simple memory channel performs better than the original Viterbi algorithm with the same decoding complexity.

The M-state Markov model was reviewed in this report. The process of identifying the parameters of the M-state model from the coefficients A_i and $A_i(n_i, n_{i+1})$ of the gap model was determined to be more complicated than was anticipated. An alternative, the simple partitioned Markov model was examined to determine the effect of the second order statistics, namely the interdependence of the gaps, on the error burst distribution. An alternative definition of the burst was adopted to speed up this investigation. The difference or similarity between these two definitions will be determined.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Digital Communications						
Error Correcting Codes						
Channel Modeling						

Security Classification